

Web-ohjelmointi - Harjoitustyön ohje

- Tämä dokumentti sisältää ohjeet **Web-ohjelmointi** opintojakson harjoitustyön tekemistä ja dokumentointia varten.
- [Muutamia esimerkkiharjoitustöitä](#)

Kuvaus

1. Harjoitustyössä rakennetaan opintojakson aihealueeseen liittyvä verkkosivusto, jonka fokus on käyttäliittymäohjelmoinnissa web-selaimeen JavaScriptillä ja sen eri kirjastoilla. Keskiössä on selainpäässä tapahtuva käyttäjän ja käyttäliittymän välinen vuorovaikutus. Tyypillisesti sivusto voisi ladata dataa näytettäväksi itse luomastasi tai jostakin avoimesta APIsta esim. AJAX-tekniikalla

Vaihtoehtoisesti harjoitustyössä voidaan esitellä opintojakson aihealueeseen kuuluvan [MV* frameworkin](#) käyttöönottoa/toimintaa ja demonstroida sitä **itse toteutetulla** demosovelluksella.

Lisäksi mikä tahansa opintojaksolla käsittelemätön selainpään "erikois"kirjaston käyttö osana harjoitustyötä tuo yleensä lisäarvoa. Tällaisia kirjastoja voisivat olla esim. Vue.js, Plotly.js, SvelteJS, Phaser, ...

2. Tämän opintojakson harjoitustyössä voi tarvittavin osin olla myös backendiä esim. php/mysqlä, mutta se on enemmän apuväline käyttäliittymäjutuille, joten niitä voit tehdä niin suoraviivaisesti kuin vain mahdollista. Tämän opintojakson töissä on ihan ok lukea dataa käyttäliittymään jostakin staattisestakin JSON-tiedostosta tai vastaavasta. Datan tallentamiseen voi käyttää esim. LocalStoragea. Backend-puolta ei arvioida tämän opintojakson harjoitustöissä.
3. Harjoitustyöprosessissa opiskelija oppii, soveltaa ja osoittaa opintojakson aihealueen hallintaa laajemmin ja syvemmin kuin pelkästään harjoitustehtävien avulla. Parhaimmillaan tämä näkyy sellaisten ratkaisujen soveltamisessa, jotka eivät ole suoraan nähtävissä opintojaksolla tarjotuista materiaaleista.
4. Harjoitustyön suunnittelu- ja tekoprosessi sekä tuotos dokumentoidaan HTML-muotoisella harjoitustyödokumentaatiolla.
5. Harjoitustyö tehdään optimissaan 1-3 hengen ryhmässä. Poikkeava ryhmäkoko on hyväksyttävä ennen työn aloitusta. Jokaisen työntekijän

tekemä työ on dokumentoitava selvästi.

6. Harjoitustyön aihe voidaan valita melko vapaasti, mutta aihe kannattaa esittää kurssin opettajalla ennen tekemään ryhtymistä. Opettaja joko hyväksyy, hylkää tai korjaa aihetta vastaamaan kurssin harjoitustyölle asetettavia minimivaatimuksia.
7. **Työaika** on varattu harjoitustyöhön n. **40** tuntia per henkilö (4op:n opintojakso)
8. Työ esitellään **harjoitustyöseminaarissa** erikseen sovittavan aikataulun mukaan. Esityksen voi pitää myös videoesittelynä. Verkko- ja monimuotototeutuksissa (esim. CampusOnline.fi) esittely suoritetaan aina videoesittelynä. Videoesittelyn pituus on 3-10min. Esittely on pakollinen.
9. Harjoitustyötä työstetään omalla ajalla. Lähiopetusjaksoja sisältävissä toteutuksissa ohjausta voi saada myös lähiopetuksen yhteydessä.
10. Työn **lähdekoodi** ei voi olla miltään osin salaista.
11. Pyydettyäessä opiskelijan tulee pystyä kertomaan harjoitustyössä käytettyjen ratkaisujen keskeisestä teknisestä toimintalogiikasta esim. (video)puhelun välityksellä. Harjoitustyö kaikkineen tulee olla saatavilla siihen asti, kunnes opintojakson suoritus on virallisesti hyväksytty ja kirjattu opintotietojärjestelmään.

Harjoitustyön palautus

1. Palautus: **Ks. palautuspäivämäärä kurssitoteutuksesta.**
2. Työn valmistumisesta ilmoitetaan sähköpostilla kertomalla ainoastaan URL, josta työ löytyy. Työn tulee olla tuolloin valmis ja opettajan tarkastettavissa. Mitään ZIP-paketteja ei postiteta.
3. Dokumentaation tulee sisältää selkeästi saatavilla oleva linkki videoesitykseen, jos esittely suoritetaan videona.
4. Palautuksen muotoseikoista. Yhden **URLin** takaa löytyy:
 - Työn loppudokumentaatio: Harjoitustyön nimi, tekijöiden nimet, päivämäärä ja tieto siitä, minkä kurssin harjoitustyö on kyseessä.
 - Tehtävän kuvaus
 - Käytännön toteutuksen selostus, esimerkiksi:

- Sovelluksen kokonaisrakenne mielellään graafisesti, mahdollinen luokkarakenne, funktioiden suhteet, sovelluksen suhteet tiedostoihin, mahdollisiin ulkoisiin tietovarastoihin...
- Funktiot ja luokat (JavaScript: oliot) : suositeltavaa ainakin tärkeimpien toiminta dokumentoitava, välitettävät parametrit
- Ulkoiset tietovarastot
- Lähdekoodit - Kaikki sovelluksen lähdekoodit on oltava tarjolla
- Ajan käyttö - Selvitys työhön käytetystä ajasta ja tekijöiden roolista harjoitustyön tekemisessä sekä työn jakaantumisesta ryhmän sisällä.
- Itsearvio – oma arvio työstä muutamalla virkkeellä.
Arvosanaehdotus/tekijä. Onnistumiset, puutteet, huomioita. Opintojakson arvioiva opettaja ottaa lopullisessa arvioinnissa myös huomioon sen kuinka realistinen oma arvosanaehdotus on.

4. Linkki kaiken edellisen sisältävään **zip**-pakettiin.

Harjoitustyön tulee siis olla kokonaisuudessaan verkossa tarkastettavissa yhden URL-osoitteen takana. Suosituksena on, että työ palautetaan *student.labranet.jamk.fi*-palvelimella tai ainakin lopullinen arvosteltavaksi jätettävä versio julkaistaan siellä. Mahdollista on myös käyttää jotakin omaa palvelinta.

Arviointiin vaikuttaa

- 65% Toteutetun sivuston lopputulos: Käytetyn tekniikkakirjon laajuus ja laatu (laajempi sisältö antaa mahdollisuuksia laajemman kirjon esittelylle), web-käyttöliittymän ulkoasu ja käytettävyys
- 25% Harjoitustyödokumentaatio: laajuus ja laatu
- 10% Työn esittely
- Myös prosessia arvioidaan osana edellä mainittuja kohtia

Erästä luonnostelua mahdolliseksi harjoitustyöksi

Laajenna harjoituksen <http://netisto.fi/ttms0500-s2019/harjoitukset/harjoitukset4.html> tehtävää 1 vaikka sellaiseksi kiinteistövälitysyhtymän kohde-esittelysivuksi. Lisättäviä asioita voisi olla esim:

1. Lisäät paljon (~15-20kpl) erilaisia kuviteltuja kohteita JSON-tiedostoon
2. Muuta hinta-kenttä luvuksi
3. Lisää kohteille ainakin tyyppi-kenttä (okt, kerrostalo, rivitalo, ...) ja ilmoitettu-kenttä (päivämäärä) ja mahdollisesti muita kenttiä

Varsinainen työ voisi sisältää joitakin seuraavia ominaisuuksia:

1. Tee sivusta käyttöliittymältään oman kädenjälkesi näköinen
2. Päänäkymässä ei ehkä näkyisi kuin viimeksi lisätty kohde (sen aikaleiman perusteella)
3. Käyttäjä voisi hakea kohteita rajattuna hinnan, tyytin, perusteella. Saat itse miettiä pitäisikö käyttäjän kirjoittaa nuo arvot käyttöliittymään vai valitsisiko käyttäjä ne esim. pudotusvalikosta vai miten tulisi toimia
4. Käyttäjä voisi lajitella kohteita hinnan mukaan
5. Käyttäjä voisi valita, mitä tietoja kohteista näytetään (esim. rastittamalla checkboxeista)
6. Käyttäjä voisi tallentaa kohteen itselleen "kiinnostavaksi kohteeksi" esim selaimen localStorageiin

Laajemmasta työstä voisi löytyä vielä:

1. Kohteiden osoitteiden perusteella käyttäjän olisi mahdollista saada kohteet näkyviin esim. kartallekin jonkinlaisilla markkereilla merkittynä ja markkeria klikatessa kohteesta saataisiin näkyville ne muutkin tiedot
2. ks. lisää ideoita esim etuovi.com:sta

Jätetty tarkoituksella tyhjäksi